

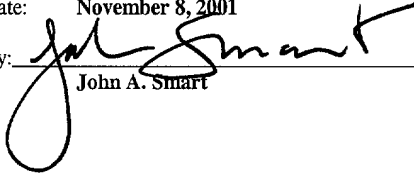
I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated below and is addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Docket No. **LS/0008.01**

"Express Mail" label number: EF414686908US

Date: **November 8, 2001**

By:


John A. Smart

PATENT APPLICATION

WAVELET TRANSFORMATION ENGINE

Inventor: MARK J. SANDFORD, a citizen of The United States residing in Lake Forrest, CA.

Assignee: LightSurf Technologies, Inc.

John A. Smart
Reg. No. 34,929

WAVELET TRANSFORMATION ENGINE

RELATED APPLICATIONS

5 The present application is related to and claims the benefit of priority of the following commonly-owned provisional application(s): application serial no. 60/262,568 (Docket No. LS/0008.00), filed January 18, 2001, entitled "Wavelet Transformation Engine", of which the present application is a non-provisional application thereof. The disclosure of the foregoing application is hereby incorporated by reference in its entirety, including any appendices or
10 attachments thereof, for all purposes.

COPYRIGHT NOTICE

 A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and
15 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

1. Field of the Invention

 The present invention relates generally to digital image processing and, more
20 particularly, to wavelet-based compression of digital images.

2. Description of the Background Art

 Today, digital imaging, particularly in the form of digital cameras, is a prevalent reality that affords a new way to capture photos using a solid-state image sensor instead of
25 traditional film. A digital camera functions by recording incoming light on some sort of sensing mechanisms and then processes that information (basically, through analog-to-digital conversion) to create a memory image of the target picture. A digital camera's biggest advantage is that it creates images digitally thus making it easy to transfer images between all

kinds of devices and applications. For instance, one can easily insert digital images into word processing documents, send them by e-mail to friends, or post them on a Web site where anyone in the world can see them. Additionally, one can use photo-editing software to manipulate digital images to improve or alter them. For example, one can crop them, remove red-eye, change colors or contrast, and even add and delete elements. Digital cameras also provide immediate access to one's images, thus avoiding the hassle and delay of film processing. All told, digital photography is becoming increasingly popular because of the flexibility it gives the user when he or she wants to use or distribute an image.

In order to generate an image of quality that is roughly comparable to a conventional photograph, a substantial amount of information must be captured and processed. For example, a low-resolution 640 x 480 image has 307,200 pixels. If each pixel uses 24 bits (3 bytes) for true color, a single image takes up about a megabyte of storage space. As the resolution increases, so does the image's file size. At a resolution of 1024 x 768, each 24-bit picture takes up 2.5 megabytes. Because of the large size of this information, digital cameras usually do not store a picture in its raw digital format but, instead, apply compression technique to the image so that it can be stored in a standard compressed image format, such as JPEG (Joint Photographic Experts Group). Compressing images allows the user to save more images on the camera's "digital film," such as flash memory (available in a variety of specific formats) or other facsimile of film. It also allows the user to download and display those images more quickly.

Wavelet-based compression is the newest compression technology available on the consumer market. Wavelet technology enables digital images and video to be compressed by removing all obvious redundancy and using only the areas, which can be perceived, by the human eye primarily edges and shading changes which are generally represented by high frequencies. Wavelet technology filters the entire field or each frame at as a single entity. This approach results in smother images, as opposed to traditional JPEG style processing that may yield blocky images due to its block oriented processing. As a result, the technique provides a significant advantage for still images over the more traditional DCT-based methods that are used in the JPEG (baseline) industry standard. All told, wavelet-based methods offer the advantage of a better trade-off between complexity, compression, and

quality. Accordingly, wavelet-based techniques are expanding in the field of still image and video compression at an ever-increasing rate.

The basic concept behind wavelets is that, rather than performing a tiling approach (i.e., breaking an image down into small segments), filters (and sub-band coding) are applied over the entirety of an image. This is illustrated in Fig. 1. First, a high pass filter and low pass filter are applied in parallel to separate the image, such that two results are generated one being the high-pass filtered results and low-pass filtered results, as shown at (a). Using Nyquist sampling theory, as the resultant images have reduced bandwidth, such that only half the amount of data is required for complete frequency representation, and thus the image can be sub-sampled by a factor of two with no information loss in the direction of filtering. This sub-sampling is simply done by removing every other resultant sample. This is done to both the high-pass and low pass results, such that the resultant data size is the same as the original image size. This approach is performed first in one direction either horizontal or vertical and then repeated in the other direction in a manner to produce four quadrants: first along an image's horizontal axis to produce high-pass and low-pass filtered halves, then repeating along an image's vertical access to produce high-pass and low-pass filtered quadrants, as shown at (b). Here, the upper left quadrant (Q1) represents low-pass horizontal and vertical image data. The upper right quadrant (Q2) represents high-pass horizontal and low-pass vertical image data. The lower left quadrant (Q3) represents low-pass horizontal and high-pass vertical image data. The lower right quadrant (Q4) represents high-pass horizontal and vertical image data. This process can be repeated at multiple levels, as shown at (c) - (e), by repeating the process on each resultant Q1 result each being a quarter the size of the previous level. This may continue until the resultant Q1 block is too small to continue further, yielding best compression.

Further description of the wavelet-based compression may be found, for instance, in the technical and trade literature. See e.g., Pigeon, S., *Image Compression with Wavelets*, Dr. Dobb's Journal, August 1999, pp. 111-115. The disclosure of the foregoing is hereby incorporated by reference, for all purposes.

Historically, wavelet processing has been implemented in software. In user-operated imaging devices, however, responsiveness to users is paramount. Therefore, there is great interest in finding a wavelet-based compression technique that is optimized for a given

hardware environment. A particular problem arises when attempting to do hardware-based wavelet processing due to existing memory architecture, however. In particular, applying vertical filters using existing memory architecture is problematic.

Current memory architecture in widespread use (e.g., synchronous DRAMs --

5 SDRAMs) is optimized for sequential data access in a horizontal manner, such as page-based or row-based access. For example, in the SDRAM memory commonly employed in PCs, horizontal access may be achieved on the order of 7-10 nanoseconds. This speed results from a pre-fetch pipelining mechanism, which is optimized for fetching the next data element (e.g., machine word) in a given row ("page"). Vertical access (e.g., accessing a pixel value
10 below), in contrast, requires around 120 nanoseconds, a ten-fold increase in access cost. This increased cost results from the time-intensive task of switching to another row of memory cells. Here, the underlying memory access mechanism must be reconfigured to switch to the next memory page 2 to access the next group of bits.

One approach to mitigating the above limitation of current memory architecture is to
15 employ alternative memory architecture -- that is, forego use of RAM that is page oriented. One such example is static RAM (SRAM). Unfortunately, that approach has distinct disadvantages in terms of greatly increased cost, power requirements, and larger chip size. It is instead advantageous to find a solution that may be implemented using less-costly page-based memory architecture, if such a solution is possible.

20 All told, in a hardware-implemented wavelet processing approach, memory access becomes a limiting factor to a cost-effective solution. Therefore, there is great interest in finding a hardware-implemented wavelet-based compression solution that may be implemented in less-costly, page-based memory architecture (e.g., SDRAM), and do so in a manner that overcomes the inherent speed disadvantage encountered due to the horizontal-
25 optimized access strategy employed by page-based memory architectures.

GLOSSARY

The following definitions, which are provided for purposes of illustration not limitation, may assist in understanding the detailed discussion that follows.

5 *ASIC*: Short for Application Specific Integrated Circuit, a chip designed for a particular application. ASICs are built by connecting existing circuit building blocks in new ways. Since the building blocks already exist in a library, it is much easier to produce a new ASIC than to design a new chip from scratch.

10 *JPEG*: Short for Joint Photographic Experts Group, JPEG is a lossy compression technique for color images. Although it can reduce files sizes to about 5% of their normal size, some detail is lost in the compression. See e.g., Nelson, M. et al., *The Data Compression Book*, Second Edition, Chapter 11: Lossy Graphics Compression (particularly at pp. 326-330), M&T Books, 1996.

15 *wavelet*: A mathematical function used in compressing images. Images compressed using wavelets are smaller than JPEG images and can be transferred and downloaded at quicker speeds. See e.g., Pigeon, S., *Image Compression with Wavelets*, Dr. Dobb's Journal, August 1999, pp. 111-115.

SUMMARY OF THE INVENTION

An ASIC-implemented wavelet transformation engine (circuit) providing a wavelet filter is described. The wavelet filter itself provides up to a 9-stage FIR (finite impulse response) filter with symmetrical coefficients. The architecture of the filter includes data inputs, a bank of shift registers (register bank), coefficient registers, a multiplier/accumulator, a sub-sampling component, and output (results) registers. The design employs multiplexors for controlling inputs to the coefficient registers and output (results) registers.

The data inputs, which include high-pass inputs and low-pass inputs, feed into the register bank. In the currently preferred embodiment, a nine-tap filter is implemented, thus requiring that the register bank include nine registers for storing nine incoming data points. These data points or values are to be multiplied against nine coefficients, which are stored at coefficient registers. Two different sets of coefficients are used to do high- and low-pass filtering. With this configuration, a series of inputted data are shifted across/against a set of coefficients, which implement specific filter characteristics. The embodiment is fully programmable, so a variety of other wavelet filters up to nine-taps (symmetric or not) may be implemented.

In the currently preferred embodiment, the wavelet filter is configured to perform as a FBI 7-9 wavelet filter with zeros inserted in unused coefficient locations. As a simplification and performance enhancement technique, rather than filtering an entire row for both high-pass and low-pass filters and then dropping alternate results (as is possible from the Nyquist theorem), the low pass and high-pass filters are alternated such that only results are generated for either low-pass or high-pass, reducing the filter processing time by a half yet still yielding the full informational content of the underlying digital image.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating wavelet-based image compression.

Figs. 2A-B are schematic diagrams illustrating a preferred embodiment and an alternative embodiment of an ASIC-implemented wavelet transformation engine (circuit) of the present invention.

Figs. 3A-B present block diagrams illustrating a mirroring function of the filter, in which the data at each end of a line of data being filtered is mirrored.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The following description will focus on the presently preferred embodiment of the present invention, which may be implemented in a low-cost ASIC (application-specific integrated circuit) chip. The present invention, however, is not limited to just ASIC-based implementations. Instead, those skilled in the art will find that the present invention may be advantageously embodied in other environments, including, for example, a field programmable gate array (FPGA) chip. Therefore, the description of the exemplary embodiments that follows is for purposes of illustration and not limitation.

I. ASIC-based implementation

The present invention may be implemented on an ASIC. An ASIC is an integrated circuit or "chip" that has been built for a specific application. Integrated circuits are traditionally designed with general-purpose functions that allow designers to design systems in the form of integrated circuit boards by connecting integrated circuits with selected functions to one another. For example, most integrated circuits have general functions, such as combinational logic, shift registers, and the like, and are connected to one another on circuit boards. Designers may use ASIC to consolidate many integrated circuits into a single package thereby reducing circuit board size requirements and power consumption. An ASIC implements custom functionality according to a description, which is provided in an abstract technology-independent fashion for instance using a Hardware Description Language (HDL), such as VHDL (Very High Speed Integrated Circuit Hardware Description Language) or Verilog Hardware Description Language.

ASICs may incorporate programmable logic arrays, field programmable gate arrays, cell based devices, and fully custom designed devices. ASICs may include such general function circuits that are connected to perform specific applications as systems, such as, a disk controller, a communications protocol, a bus interface, a voice coder, and the like. An ASIC may include on a single integrated circuit the circuitry that is typically built on a circuit board. ASIC devices are available from a variety of suppliers, including Fujitsu, Hyundai Electronics America, and Texas Instruments.

The use of an ASIC-based implementation is presented for purposes of illustrating the basic underlying architecture and operation of the present invention. An ASIC-based implementation is not necessary to the invention, but is used to provide a framework for discussion. Instead, the present invention may be implemented in any type of circuitry capable of supporting the processes of the present invention presented in detail below.

II. Implementation of a wavelet filter using low-cost memory

A. Design

1. Basic architecture

Fig. 2A is a schematic diagram showing an ASIC-implemented wavelet transformation engine (circuit) providing a wavelet filter 200 that operates under control of a DSP (digital signal processing) circuit 290, which includes or controls a clock providing a clock tick at a specified time interval. As shown, the ASIC 200 includes data inputs 210, bank of shift registers (register bank) 220, multiplexor set 230, coefficient (registers) 240, multiplier/accumulator circuit 250, sub-sampling component 260, multiplexor 270, and output (results) registers 271, 273. In a clock-synchronized fashion, the DSP circuit 290 coordinates operation of the components.

The detailed design of the engine's filter 200 is as follows. The wavelet filter 200 provides up to a 9-stage FIR (finite impulse response) filter with coefficients that can be symmetrical or nonsymmetrical as desired. At a particular clock interval, successive pixels in the DSP-controlled image memory are shifted (e.g., horizontally, for application of a horizontal filter) into the register bank 220. In this manner, the register bank 220, at any given time, is employed to provide a neighborhood of pixel values for a particular pixel from the underlying digital image. Here, in a clock-synchronized fashion, pixel values (from the current neighborhood under exam) are copied into the register bank 220: the data inputs 210, which include high-pass inputs 211 (WT_HP_IN_1 through WT_HP_IN_8) and low-pass inputs 213 (WT_LP_IN_1 through WT_LP_IN_8), feed into the register bank 220.

In the currently preferred embodiments, the DSP 290 itself may be Inicore's iniDSP or other similar processors from other various DSP vendors (e.g., Fujitsu, Hyundai, Texas Instruments, or the like). Further, the DSP can also be replaced by a general-purpose

processor (e.g., Intel-based or Motorola-based), or even a DMA engine. DSP's are preferred as in general they contain DAG (data address generation) units that are optimized for this type of (e.g., image-based) data movement yet remain programmable. General purpose processors, on the other hand, might be preferred in low cost applications in which a DSP
5 does not provide enough flexibility (and two processor would not be cost effective). A "hardwired" DMA engine may provide the fastest implementation, but has the disadvantage of lack of programmability.

In the currently preferred embodiment, a nine-tap filter is implemented, thus requiring that the register bank 210 include nine registers for storing nine incoming data points, as is
10 illustrated in the figure. These data points or values are to be multiplied against nine coefficients (pixel weightings), which are stored at coefficient registers 240. Two different sets of coefficients are used to do high-pass filtering and low-pass. Assuming symmetrical coefficients each set need only store five values for populating the nine coefficient registers 240: WP_HP_COEFF0 through WP_HP_COEFF4 for high pass filtering, and
15 WP_LP_COEFF0 through WP_LP_COEFF4 for low pass filtering. With the above configuration and under control of DSP 290, a series of inputted data are shifted across/against a set of coefficients (stored at 240), which implement specific filter characteristics. The embodiment is fully programmable, so a variety of other wavelet filters up to nine-taps (symmetric or not) may be implemented.

In the currently preferred embodiment, the wavelet filter 200 is configured to perform as a FBI 7-9 wavelet filter with zeros inserted in unused coefficient locations. As a simplification and performance enhancement technique, rather than filtering an entire row for both high-pass and low-pass filters and then dropping alternate results, as is possible from the Nyquist theorem (as mention in the wavelet filter description), the low pass and high-pass
20 filters are alternated such that only results are generated for either low-pass or high-pass, reducing the filter processing time by a half yet still yielding the full informational content of the digital image.

To increase performance for column processing, an 8-stage pipelined filter is actually used (8 parallel filters are implemented) at each register position (within register bank 220),
30 thus allowing the 9-stage filter 200 to in fact process eight image columns in parallel (pipelined). The background shadows (e.g., shown at 221) represent eight lines that are to be

processed in parallel. Thus, at a given instance, the filter selects which line of data is to be processed at a given instance. This selection is effected using the multiplexor set 230, which operates under control of the DSP 290 (via MUX control line 235). The lowest three address bits are used to select which filter is being used. The filter with the lowest three address bits
5 equal to "000" has special mirroring features described below.

The following table summarizes the functionality of each of the foregoing registers and accompanying support registers (WT_SATURATION and WT_CTRL, described below).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

TABLE 1

Register	Size	Access	Description
WT_HP_IN_1-8	15:2	W	High pass filter inputs 1-8 (left aligned)
WT_LP_IN_1-8	15:2	W	Low pass filter inputs 1-8 (left aligned)
WT_HP_RESULT	16	R	High pass filter result
WT_LP_RESULT	16	R	Low pass filter result
WT_SATURATION (These two flags are sticky bits; one they are set because of saturation, they will remain set until they are cleared by software.)	[0]	R	High pass saturation flag '0': no saturation '1': clear high pass saturation flag
		W	Clear high pass saturation flag '0': no action '1': clear high pass saturation flag
	[1]	R	High pass saturation flag '0': no saturation '1': clear high pass saturation flag
		W	Clear high pass saturation flag '0': no action '1': clear high pass saturation flag
WT_HP_COEFF0	15:2	R/W	high-pass coefficient 0 (1:0 => "00")
WT_HP_COEFF1	15:2	R/W	high-pass coefficient 1 (1:0 => "00")
WT_HP_COEFF2	15:2	R/W	high-pass coefficient 2 (1:0 => "00")
WT_HP_COEFF3	15:2	R/W	high-pass coefficient 3 (1:0 => "00")
WT_HP_COEFF4	15:2	R/W	high-pass coefficient 4 (1:0 => "00")
WT_LP_COEFF0	15:2	R/W	low-pass coefficient 0 (1:0 => "00")
WT_LP_COEFF1	15:2	R/W	low-pass coefficient 1 (1:0 => "00")
WT_LP_COEFF2	15:2	R/W	low-pass coefficient 2 (1:0 => "00")
WT_LP_COEFF3	15:2	R/W	low-pass coefficient 3 (1:0 => "00")
WT_LP_COEFF4	15:2	R/W	low-pass coefficient 4 (1:0 => "00")
WT_CTRL	[0]	W	mirror start '0': no action '1': the next write to WT_HP_IN_1 or WT_LP_IN_1 mirrors the oldest 4 values and clears this bit
	[1]	W	mirror end '0': no action '1': mirror the last 4 values
	[2]	W	shift end with high pass coefficients '0': no action '1': shift one end value into filter
	[3]	W	shift end with low pass coefficients '0': no action '1': shift one end value into filter

2. Alternative embodiment

Since the filter is often symmetric (i.e., symmetrical about center coefficient **c4**), the present invention may be implemented in the alternative embodiment illustrated in Fig. 2B. More particularly, in the alternative wavelet filter embodiment 200a, the symmetry of the filter is used to reduce the hardware required and thus the area (cost) and power consumption of the chip. By knowing that given pairs of coefficients (other than the center coefficient) are the same, the alternative embodiment may employ a single multiplier for that corresponding coefficient after summing the input data, resulting in a mathematically equivalent operation. Thus, the embodiment is modified such that the output of eight of the nine registers bank (register bank 220a) is fed, via multiplexor circuitry 230a, to accumulators 231; the number of required coefficient registers (now shown at 240a) is reduced from nine to five.

As two of the values to be filtered will be multiplied by the same number, one can use the communicative property of multiplication to add the two numbers and then perform a single multiply:

$$A*N + B*N = (A+B)*N$$

The multipliers require by far the largest number of gates and thus power and area. Thus, the reduced version will be close to 5/9ths the size of the full version (the center coefficient remains). The reduced version is slower in operation as there is an additional addition operation added to the pipeline, but the reduced size and width of operations counteract this limitation.

All told, the reduced alternative version is more cost effective and power efficient. However, it limits the application to symmetric wavelet filters. Using the full nine coefficient architecture, on the other hand, allows for maximum flexibility by being able to use any symmetric or non-symmetric for either wavelet processing or more general FIR filter processing, which may be useful in other operations within modern digital cameras. The full version also allows for increased speed relative to the reduced version due to the fewer operations in the processing pipeline.

3. Mirroring features

Although FIR filters assume that the available data is continuous, this is not a reasonable assumption in most applications. That is especially the case for image processing applications in which each row or column of data is processed individually and thus provides two end conditions which must be constrained to control the filter response. Multiple possible mechanisms exist to constrain the end conditions. The one selected in the currently preferred embodiment is mirroring in which the data at each end of the line of data being filtered is mirrored. This mechanism both controls the filter response and guarantees that the image can be accurately reconstructed. There is an additional benefit for performance as this allows this repeated data to be read only once saving valuable processing time.

As illustrated in Figs. 3A-B, this is accomplished by a hardware data copy mechanism that repeats the appropriate data elements to implement a “mirroring” feature. In Fig. 3A, the nine blocks in the figure represents the 9-stage filter with the lowest three address bits equal to “000”, either WT_HP_IN_1 or WT_LP_IN_1. Here, after the first four elements are placed into the filter, a *mirror_start* function is initiated. This is specified by values placed in a 4-bit control register (WT_CTRL), as illustrated in Table 1. A “mirror start” bit 301 is set after the first four values have been filled into the filter. The mirror happens when writing value #4 into the filter. This causes the four data elements preceding the edge of the row to be copied in reverse order such that the filter receives symmetric data centered on the edge of the image.

Fig. 3B illustrates a *mirror_end* function, which is used in combination with a *shift_end* function. The *mirror_end* function is invoked after the last value has been filled into the filter. This is followed by invoking the *shift_end* function four times to shift in the copy values into the filter. To apply this mirror feature to column processing as well, one can simply read the columns in the desired order; no explicit mirroring is needed.

B. Image filtering operation

Basic operation of the wavelet filter of the present invention will be described by illustrating specific operational steps employed by the wavelet filter 200 for filtering images. The specific steps of the process, which operate under the timing control of the DSP chip 290, are as follows. During use of the filter, a target image exists in external memory (e.g., SDRAM). It is read in, in a burst of eight consecutive pixels, in a horizontal line for

application of a horizontal filter. The pixels are brought into memory that is accessible to the DSP chip 290. Given the nine registers across the top, as the filter walks across the line of image data (i.e., receives input comprising successive lines of image data), the filter shifts data in from one side of the register bank 220 to the other side (e.g., from left to right for the embodiment shown in Fig. 2A), so that the register bank 220 stores a sequence of pixels from image memory (e.g., taken from successive horizontal lines of the image, for application of a horizontal filter).

The first data element (i.e., first word pixel, which is 16 bits in size) is written into the low pass zero register. Next, the low pass filtering coefficients are applied, and a low pass result is generated. The second data element is written into the high pass zero register. Similarly, filtering occurs using the high-pass coefficients, with a high pass result being generated. Thus during operation, the filter alternates between high pass and low pass. Here, the data shifts by one (pixel value) but alternates between using high pass or low pass coefficients. In the case of vertical filtering (i.e., filtering pixels where neighbors are located vertically), the eight pixels from the first row are placed in the low-pass registers. The second row is placed in the high-pass registers. In other words, odd number rows are placed in the low pass registers; even number rows are placed in the high pass registers.

As previously described, the coefficients set 240 stores the nine coefficients that are applied against the image data values held by the register bank 220. In the currently preferred embodiment, the filter is symmetric. Thus, the first and last coefficients store the same value (shown as **c0**), the second and eighth coefficients store the same value (shown as **c1**), and so forth and so on, with the center coefficient (shown as **c4**) being the only unique value. The filter alternates such that the odd pixel values (i.e., 1st, 3rd, 5th and 7th values) are placed in the low pass registers, with the even the pixel values being placed in the high pass registers. The coefficient set 240 is multiplied/accumulated against the data set using the multiplier/accumulator circuit 250. Based on the pixel values from the supplied pixel neighborhood and based on the coefficient weightings, the multiplier/adder circuit generates a new pixel value. This resulting data is, in turn, saturated down to a 16-bit value (i.e., taking the most significant 16 bits), as shown by the sub-sampling component 260. The operation of the sub-sampling component 260 (e.g., enabling and disabling) is configurable by setting specific flags in the WT_SATURATION register (listed in Table 1).

Using multiplexor 270, the resulting output (i.e., newest result data element) is returned as a low pass result in low-pass result register 273 (WT_LP_RESULT) when the filter is processing a low pass filter, or is returned as a high-pass result in high-pass result register 271 (WT_HP_RESULT) when the filter is processing a high pass filter. Operation of the multiplexor 270 is under control of the DSP 290, via MUX control line 275. The foregoing processing is repeated for all of the pixels of the underlying image for rendering an image-processed (i.e., wavelet transformed) version of that image.

While the invention is described in some detail with specific reference to a single-preferred embodiment and certain alternatives, there is no intent to limit the invention to that particular embodiment or those specific alternatives. For instance, those skilled in the art will appreciate that modifications may be made to the preferred embodiment without departing from the teachings of the present invention.